

What is claimed is:

1. An apparatus for generating enterprise java beans based on a class diagram, comprising:

5 a class diagram receiving means for receiving the class diagram;

an inheritance relationship removing means for eliminating an inheritance relationship existing among classes in the class diagram;

10 an enterprise java bean extracting means for extracting the enterprise java beans based on the class diagram whose inheritance relationship was eliminated;

an enterprise java bean interface generating means for producing a remote interface and a home interface of the enterprise java beans extracted by the enterprise java bean extracting means;

15 an enterprise java bean grouping means for grouping enterprise java beans related to each other whose interfaces are produced by the enterprise java bean interface generating means; and

20 a façade pattern applying means for applying a façade pattern to the enterprise java beans grouped by the enterprise java bean grouping means to thereby unify external interfaces.

25 2. The apparatus of claim 1, wherein the enterprise java bean extracting means maps a class, which is stored in a database and should maintain its persistence, to an entity

bean, and a class whose persistence needs not to be guaranteed to a session bean.

3. The apparatus of claim 1, wherein the enterprise java bean interface generating means produces a primary key if a class is mapped to an entity bean.

4. A method for generating enterprise java beans based on a class diagram, comprising the steps of:

(a) eliminating an inheritance relationship existing among classes in the class diagram;

(b) extracting the enterprise java beans based on classes whose inheritance relationship is eliminated;

(c) adding an enterprise java bean interface to the extracted enterprise java beans;

(d) grouping interface added enterprise java beans which are related to each other; and

(e) applying a façade pattern to the grouped enterprise java beans.

5. The method as recited in claim 4, wherein the step (b) includes the steps of:

(b1) extracting classes from the class diagram and determining whether or not generating the extracted classes to the enterprise java beans;

(b2) if it is determined that the extracted classes are not generated to the enterprise java beans, returning to the

step (b1) and, if otherwise, extracting classes to be generated to the enterprise java beans;

(b3) deciding if the persistence of the extracted classes is guaranteed;

5 (b4) if the extracted classes have the persistence, detecting classes having the persistence and generating the detected classes to the enterprise java beans; and

(b5) if the extracted classes do not have the persistence, determining classes not having the persistence and producing the determined classes to session beans.

6. The method as recited in claim 5, wherein the step (b4) includes the steps of:

(b41) extracting the classes having the persistence;

15 (b42) deciding whether said extraction is either property extraction to be stored in a database or operation extraction;

(b43) if said extraction is determined as the property extraction, generating an entity bean by extracting a property to be stored in the database; and

20 (b44) if said extraction is decided as the operation extraction, producing a session bean by extracting an operation.

25 7. The method as recited in claim 4, wherein the step (c) includes the steps of:

(c1) taking out business logic operations related to a problem region from an extracted empty class;

(c2) adding the extracted business logic operations to a remote interface;

(c3) adding operations related to the enterprise java bean generation and detection to a home interface; and

5 (c4) identifying the extracted enterprise java bean as an entity bean and adding a primary key class.

8. The method as recited in claim 7, wherein the step (d) includes the steps of:

10 (d1) determining whether there is a class relationship;

(d2) if it is determined that there is the class relationship, extracting the class relationship and deciding if the class relationship is an inheritance relationship;

15 (d3) if it is decided that the class relationship is the inheritance relationship, grouping an upper class and a lower class which establish the inheritance relationship with each other and proceeding to the step (d1), and, if otherwise, checking out whether the class relationship is an aggregation relationship or not;

20 (d4) if the class relationship is determined as the aggregation relationship, grouping a whole class and a part class which establish the aggregation relationship with each other and then returning to the step (d1), and, if otherwise, going to the step (d1).

25 9. A computer program product for use in an enterprise java bean generating apparatus including a mass storage

processor, comprising:

a computer readable medium;

first program instruction means for eliminating an inheritance relationship existing among classes in the class diagram;

second program instruction means for extracting enterprise java beans based on classes whose inheritance relationship is eliminated;

third program instruction means for adding an enterprise java bean interface to the extracted enterprise java beans;

fourth program instruction means for grouping interface added enterprise java beans which are related to each other; and

fifth program instruction means for applying a façade pattern to the grouped enterprise java beans.